An improved meshless Shepard and least squares method possessing the delta property and requiring no singular weight function

Xiaoying Zhuang[1,3], Hehua Zhu[1*] and Charles Augarde[2]

1 Department of Geotechnical Engineering, Tongji University, Shanghai, 200092, China.

2 School of Engineering and Computing Sciences, South Road, Durham University, DH1 3L3, UK.

3 School of Civil and Resource Engineering, The University of Western Australia, M051, 35 Stirling Highway, Crawley, WA 6009, Australia

[*]Corresponding author.

E-mail: zhuhehua@tongji.edu.cn, Tel.: +86-21-65983982, Fax: +86-21-65985140

Abstract

The meshless Shepard and least squares (MSLS) method and the meshless Shepard (MS) method are Partition of Unity (PU) based meshless interpolations which eliminate the problems by other meshless methods such as the difficulty in direct imposition of the essential boundary conditions. However, singular weight functions have to be used in both methods to enforce the approximation interpolatory, which leads to the loss of smoothness in approximation and locally oscillatory results. In this paper, an improved MSLS interpolation is developed by using dually defined nodal supports such that no singular weight function is required. The proposed interpolation satisfies the delta property at boundary nodes and the compatibility condition throughout the domain, and is capable of exactly reproducing the basis function. The computational cost of the present interpolation is much lower than the Moving Least-Squares (MLS) approximation which is probably the most widely used meshless interpolation at present.

Keywords: *Meshless; Shepard shape function; Partition of unity; Delta property; Compatibility*

1. Introduction

In the past decade, meshless methods have benefited from much theoretical development and engineering application, since they offer the possibility of a discretised approach without meshing, a major overhead in the finite element method

(FEM). A wide variety of meshless methods have been proposed as outlined in recent surveys [1-3]. Remarkable successes have been reported in applying these methods for analyzing challenging engineering problems, namely, fracture modelling [4-6], plate problems [7], finite deformation problem [8,9], consolidation problem [10], dynamic simulation [11], three-dimensional problems [12,13], topology-optimization of structures and thermodynamic analysis, where laborious preprocessing involved in the FEM is avoided.

As concluded in [3] the difference between the various meshless methods is in the type of approximations used in obtaining the shape functions. Some widely used meshless approximations are the moving least-squares (MLS) approximation, Shepard shape functions, partition of unity (PU), radial basis functions (RBF), reproducing kernel particle approximation (RKPA) [14,15], point interpolation (PI) and Kriging interpolation (KI) and a generalized meshless approximation [16]. The MLS approximation [17] is probably the most widely used meshless approximation at present due to its advantages of field continuity in a global sense, completeness of approximation and robustness of calculation results. However, the MLS approximation suffers from a number of problems that practically limit its application, namely the high computational cost in obtaining the shape functions and also their derivatives, the retention of accuracy with respect to nodal arrangement and the difficulty with which essential boundary conditions can be imposed due to the lack of the Kronecker delta property. Efforts have been made to address these problems by various means in the past. In [18], explicit expressions are proposed for computing shape functions and diffuse derivatives of shape functions by assuming some terms constant and complete derivatives of shape functions. However, these formulations are restricted to certain nodal arrangements and have to be derived separately when the number of nodes in support changes, and the formulation grows unwieldy when there are a large number of nodes in support. In [19], the use of the orthogonal basis function in the element-free Galerkin (EFG) method is investigated in terms of the solution accuracy and nodal arrangement. To remove the difficulty in imposing the boundary conditions, singular weight functions are introduced in [20] to produce an interpolatory MLS approximation. In [21], a method for direct imposition of essential boundary conditions is proposed to reform the global stiffness matrix by using a transformation matrix to enforce boundary nodes taking nodal values. All these above describe efforts that help to alleviate the

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

problems, however none are capable of dealing with the problems satisfactorily without the loss of generality in the formulation.

On the other hand, researchers have also started to explore the possibility of new meshless interpolations using Shepard shape functions, the lowest order form of the MLS shape functions. Unfortunately, the results are of low accuracy if Shepard shape functions are directly used because they have only zeroth order continuity. There have been some efforts devoted to the construction of high order consistent interpolation using Shepard shape functions. For example, a consistent pseudo-derivative is proposed in [22] which can preserve the linear consistency of interpolation approximation by linearly combining the derivatives of Shepard functions together. In [23], an octree partition of unity method was developed by using the data structures of octrees and Shepard shape functions as a PU. Griebel and Schweitzer [24-26] proposed a particle-PU method by employing a localized version of Shepard's method. These methods are generally more efficient than some existing meshless methods, and show a high rate of convergence and accuracy. However, none provides a direct solution for dealing with the essential boundary conditions. In contrast, the recently developed meshless Shepard least squares (MSLS) method [27] and the meshless Shepard (MS) method [28] satisfactorily maintain the consistency of the approximations up to the order of the basis functions and also satisfy the Kronecker delta property. However, singular weight functions have to be used to enforce the shape function to be interpolatory, which results in the loss of smoothness of the interpolation and results become locally oscillatory around the node where singular weight functions are employed.

In this paper, an improved PU-based MSLS interpolation possessing the delta property without using singular weight functions is developed. The support domains at the nodes are dually defined for local approximation and for the global PU. The present interpolation is capable of exactly reproducing any function which appears in the basis.

The content of the paper is outlined as follows. In § 2, the formulation of the interpolation is described in detail including the local approximation and nodal support domain with dual definitions. The Kronecker delta property, completeness property, compatibility property, and computational efficiency of the interpolation are analyzed and discussed in § 3. The discretised formulation of the present interpolation is derived using the Galerkin weak form in § 4 followed by numerical tests demonstrating the convergence characteristics and accuracy in § 5.

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

2. Formulation of the improved MSLS interpolation

In this section, the improved meshless Shepard least squares (IMSLS) interpolation is described in detail. We start with the description of the formulation using a 2D problem domain of arbitrary geometry as shown in Fig.1. The formulation is described for the interpolation in elastostatics, with the fundamental field variable where u_I and v_I are the nodal displacements in the x and y directions respectively. The interpolation for the x-displacement at an arbitrary point $\mathbf{x} = \{x, y\}$ inside the domain is expressed as

$$u(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) u_I^l(\mathbf{x}) \tag{1}$$

where $\{\phi_I(\mathbf{x}), I = 1,...,n\}$ is a set of shape functions that forms a partition of unity, i.e. $\sum_{I=1}^{n} \phi_I(\mathbf{x}) \equiv 1$; I is the node index and n is the number of the nodes for which the supports r_{cI} include point $\mathbf{x}$ as shown in Fig.2; $u_I^l(\mathbf{x})$ here is not the nodal displacement in the FEM or the 'fictitious' nodal values in the MLS-based EFG method [1] but the local approximation at node I where the superscript l indicates local. Shepard shape functions are used as the PU given by

$$\phi_I(\mathbf{x}) = \frac{w_I(\mathbf{x})}{\sum_{J=1}^{n} w_J(\mathbf{x})}, \tag{2}$$

where $w_I(\mathbf{x})$ is the weight function of node I as in the original paper on the MSLS interpolation [27]. The construction of the IMSLS interpolation proceeds as follows: firstly, the construction of a local approximation at each node; and secondly the application of a PU approximation over the local approximation to interpolate at a point $\boldsymbol{x}$ inside the domain. The definition of nodal support and the construction of local approximations at a node will be described in detail in the following.

2.2 The local approximation at a node

The local approximation $u_I^l(\mathbf{x})$ at an arbitrary node I is given by

$$u_I^l(\mathbf{x}) = \sum_{J=1}^{M} \bar{\psi}_J^I(\mathbf{x}) u_J \tag{3}$$

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

where u_J is the nodal displacement for the J th node in support of I, M is the total number of nodes falling inside the local cover node I which is the grey circle marked with l_I in Fig.1. $\overline{\psi}_J^I(\mathbf{x})$ is given as

$$\begin{cases} \overline{\psi}_J^I(\mathbf{x}) = \psi_J^I(\mathbf{x}) - \psi_J^I(\mathbf{x}_I), & \text{for } J \neq I \\ \overline{\psi}_J^I(\mathbf{x}) = \psi_J^I(\mathbf{x}) - \psi_J^I(\mathbf{x}_I) + 1, & \text{for } J = I \end{cases}, \tag{4}$$

in which $\psi_J^I(\mathbf{x})$ is the modified least square shape function of node J at node I and is calculated by the following

$$\boldsymbol{\psi}^I(\mathbf{x}) = \left[\psi_1^I(\mathbf{x}) \quad \psi_2^I(\mathbf{x}) \quad \cdots \quad \psi_M^I(\mathbf{x})\right] = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}\mathbf{B}\,. \tag{5}$$

Here, $\mathbf{p}(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \cdots, p_m(\mathbf{x})]^{\mathrm{T}}$ is a polynomial basis, and m is the number of monomials in the basis. In the following development of the IMSLS interpolation, we will use a bilinear basis throughout in 2D, i.e. $\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = [1, x, y, xy]$. Matrices $\mathbf{A}$ and $\mathbf{B}$ in Eq. (5) are expressed as

$$\mathbf{B} = \mathbf{P}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_M \\ y_1 & y_2 & \cdots & y_M \\ x_1 y_1 & x_2 y_2 & \cdots & x_M y_M \end{bmatrix}, \tag{6}$$

and

$$\mathbf{A} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{P} \tag{7}$$

respectively. It can be seen from Eq. (4) that $\overline{\psi}_I^I(\mathbf{x}_I) = 1$, $\overline{\psi}_J^I(\mathbf{x}_I) = 0\,(J \neq I)$ and $\sum_{J=1}^{M} \overline{\psi}_J^I(\mathbf{x}) = 1$. Thus

$$u_I^l(\mathbf{x}_I) = u_I \tag{8}$$

It has been shown in [28] that if a singular weight function is used for $w_I(\mathbf{x})$ in the PU function of Eq. (2), the approximation in Eq. (1) will become interpolatory i.e. satisfying the delta property. A similar approach has been previously used by Kaljevic and Saigal [20] to make the MLS approximation interpolatory. However, the use of singular weight functions brings some problems such as the loss of smoothness in the approximation in a global sense as will be shown in the following sections.

2.3 Dual support domain of a node

The support domain of a node is the area where a node exerts influence on the field variable. In this paper it is defined as a circle centered at the node as shown in Fig. 1

although it may take any other shape such as a rectangle. Here dual support domains are defined at each node such that one is used in the construction of the local approximation and the other used in the PU approximation. In Fig. 1, for example, two support domains are associated with node I, namely a local support domain used in the local approximation, denoted as Ω_I^l with radius r_{lI}, and a PU support domain for global approximation, denoted as Ω_I with radius r_{cI}. For the construction of the local approximation, if a node falls inside the local support Ω_I^l as shown in Fig.1 (for node K), then node K will be involved in constructing the local approximation at node I. For the PU approximation, if a point say $\mathbf{x}$ in Fig.1, is contained in Ω_I, then the local approximation of the node, i.e., $u_I^l(\mathbf{x})$ will be used to approximate the field value at $\mathbf{x}$. For an arbitrary node, such as node I in Fig. 1, the size of Ω_I^l is defined by

$$r_{cI} = a \cdot b \cdot d_I , \tag{9}$$

where a is a scale factor that ranges between 1.0 and 2.0, b is a coefficient such that $b = 2$ for a node lying on the boundary and $b = 1$ for all other nodes, and d_I is the distance between I and the fifth nearest neighbour node to I. Eq. (9) is repeated for every node in the analysis. The choice of the fifth closest node is due to both the requirement of minimum nodes in support for the construction of shape functions and avoiding ill-conditioning in calculating the shape functions. When a linear basis is used, three nodes are required at least according to Eqs. (5) and (6). From our experiences, five nodes are normally sufficient for a regular nodal distribution using the linear basis.

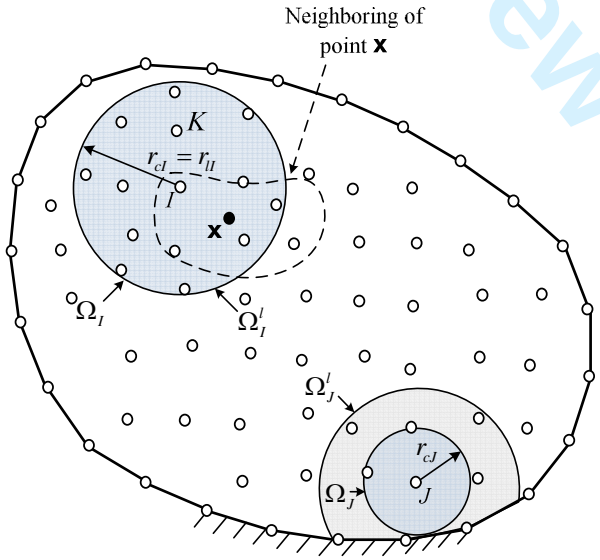


Figure 1: Dual support domains of nodes. (An interior node I with identical local and global PU support domains, and node J with differing local and global PU support domains.)

For a node having its local support domain completely inside the domain, for example the subdomain Ω_I^l of node *I* in Fig.1, the size of Ω_I is the same as Ω_I^l so that

$$r_{cI} = r_{ll} \tag{10}$$

For a node having is local support domain close to or intersecting the boundary, for example node *J* shown in Fig.1, the definition of subdomain follows these steps. Firstly, find the nearest boundary node to *J* among the neighbor nodes which belongs to Ω_I^l, and secondly calculate the distance between the nearest boundary node and *J*, denoted as d_J and then set the size of Ω_J as

$$r_{cJ} = 0.99 d_J . \tag{11}$$

If we want the approximation to take nodal values at the nodes, the size of the d_J can be taken as the distance between the *J* and its nearest node for every node *J*. In all the test examples in the paper, the following quartic spline function is used as the weight function over the support domain in Eq. (2)

$$w_I(\mathbf{x}) = \begin{cases} 1 - 6\left(\dfrac{r_I}{r_{cI}}\right)^2 + 8\left(\dfrac{r_I}{r_{cI}}\right)^3 - 3\left(\dfrac{r_I}{r_{cI}}\right)^4, & r_I \le r_{cI} \\ 0 & r_I > r_{cI} \end{cases}, \tag{12}$$

where $r_I = \|\mathbf{x} - \mathbf{x}_I\|$ is the distance between the point $\boldsymbol{x}$ and node *I*, and $\mathbf{x}_I$ is the coordinate of node *I*. For comparisons, the following singular weight function is also tested

$$w_I^{sg}(\mathbf{x}) = \left(\frac{r_I}{r_{cI}}\right)^2 \cos^2\left(\pi \frac{r_I}{r_{cI}}\right) \tag{13}$$

and where it is used in the following it is specifically pointed out. The aim of separately defining local domains and support domains is to produce IMSLS interpolations having the delta property without using a singular weight, so that the difficulties associated with the use of singular weight functions can be removed. This aim is achieved here if the domain for local approximation and domain for PU are defined by the method described above as will be proved in the following section.

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

3. Properties

3.1 Delta property at a node

Suppose essential boundary conditions are to be applied at a boundary node K and the support domains of the nodes are set according to Eqs. (10) and (11), then node K will be the only node contained in Ω_K. Thus the IMSLS interpolation in Eq. (1) at $\mathbf{x}_K$ becomes

$$u^h(\mathbf{x}_K)=\sum_{I=1}^{n}\phi_I(\mathbf{x}_K)u_I^l(\mathbf{x}_K)=\phi_K(\mathbf{x}_K)u_K^l(\mathbf{x}_K). \tag{14}$$

As there is only one node in the PU, then Eq. (2) becomes

$$\phi_K(\mathbf{x}_K)=\frac{w_K(\mathbf{x}_K)}{\sum_{J=1}^{n}w_J(\mathbf{x}_K)}=\frac{w_K(\mathbf{x}_K)}{w_K(\mathbf{x}_K)}=1 \tag{15}$$

It is known by Eq. (8) that the local approximation $u_K^l(\mathbf{x}_K)$ at node K satisfies

$$u_K^l(\mathbf{x}_K)=u_K \tag{16}$$

Substituting Eqs. (14) and (15) into (13) gives

$$u^h(\mathbf{x}_K)=u_K^l(\mathbf{x}_K)=u_K. \tag{17}$$

Hence, the present IMSLS interpolation takes nodal values at boundary nodes, and essential boundary conditions or point load conditions can be directly imposed as in the FEM.

3.2 Completeness property

The Shepard function $\phi_I(\mathbf{x})$ in Eq. (2) is the lowest order MLS shape functions and has only zeroth-order consistency, i.e. only a constant strain field can be exactly reproduced by the Shepard function. In contrast, the IMSLS interpolation in Eq. (1) is capable of exactly reproducing any function which appears in the basis of $\mathbf{p}(\mathbf{x})$ in Eq. (5). The present interpolation also preserves the order of completeness up to the order of basis function as is proved in the following. Suppose that the field over the cover of a node conforms to a given function, take the following bilinear polynomial as an example

$$\tilde{u}(x,y)=b_1+b_2x+b_3y+b_4xy. \tag{18}$$

Substituting Eq. (17) into (1) and then (2) gives

$$u_I^l(\mathbf{x}) = \sum_{J=1}^{m} \overline{\psi}_J^I(\mathbf{x})\tilde{u}(\mathbf{x}_J)$$
$$= \sum_{J=1}^{m} \psi_J^I(\mathbf{x})\tilde{u}(\mathbf{x}_J) - \sum_{J=1}^{m} \psi_J^I(\mathbf{x}_I)\tilde{u}(\mathbf{x}_J) + \tilde{u}(\mathbf{x}_I) \tag{19}$$

It has been proven in [29] that the basis function can be exactly reproduced through the least square approximation so that the first term on the *r.h.s.* of Eq. (18) becomes

$$\sum_{J=1}^{m} \psi_J^I(\mathbf{x})\tilde{u}(\mathbf{x}_J) \equiv \tilde{u}(\mathbf{x}) \tag{20}$$

and the second term becomes

$$\sum_{J=1}^{m} \psi_J^I(\mathbf{x}_I)\tilde{u}(\mathbf{x}_J) \equiv \tilde{u}(\mathbf{x}_I). \tag{21}$$

Substituting Eqs. (19) and (20) into (18) leads to

$$u_I^l(\mathbf{x}) = \tilde{u}(\mathbf{x}) - \tilde{u}(\mathbf{x}_I) + \tilde{u}(\mathbf{x}_I) = \tilde{u}(\mathbf{x}) \tag{22}$$

Substituting Eq. (21) into (1) gives

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_I(\mathbf{x}) u_I^l(\mathbf{x}) = \tilde{u}(\mathbf{x}) \cdot \sum_{i=1}^{n} \phi_I(\mathbf{x}) = \tilde{u}(\mathbf{x}) \tag{23}$$

Thus, the present IMSLS interpolation preserves completeness up to the order of the basis function.

3.3 *Compatibility*

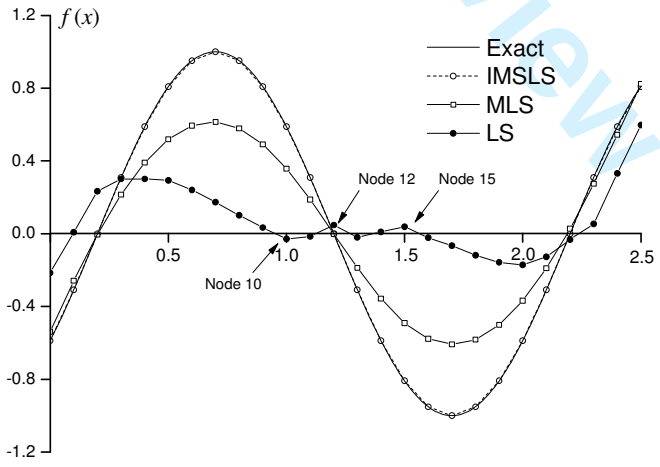
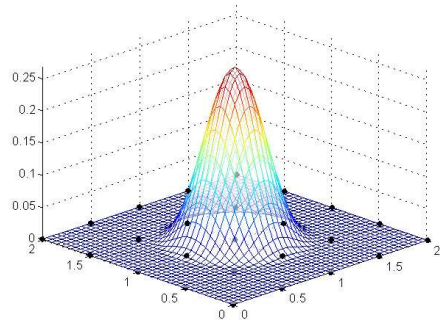


Figure 2: Curve fittings for $f(x) = \sin[(x-0.2)\pi]$ using different approximations

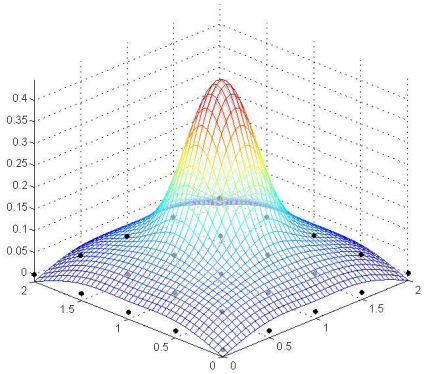
Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

In the present IMSLS interpolation, although the local cover l_I is fixed for an arbitrary node, the field function is approximated based on moving domains. Thus compatibility in the whole domain is ensured in the present IMSLS interpolation, which is the same as the MLS approximation. As an example consider, the function $f(x)=\sin[(x-0.2)\pi]$. A 1D domain ($x\in[0,\ 2.5]$) is discretised using 25 distributed nodes as shown in Fig.2, which also shows the fitting results using the LS, MLS and IMSLS approximations. It is clearly seen that the LS approximation is oscillatory and unsmooth at the region from Node 10 ($x=1.0$) to 15 ($x=1.5$). The MLS approximation is continuous in the whole domain, but cannot interpolate through all nodal values. (The readers are referred to the literature [30] for detailed discussions on compatibility for other meshless approximations). In contrast as shown by Fig. 2 the proposed IMSLS approximation is continuous and passes through nodal values.

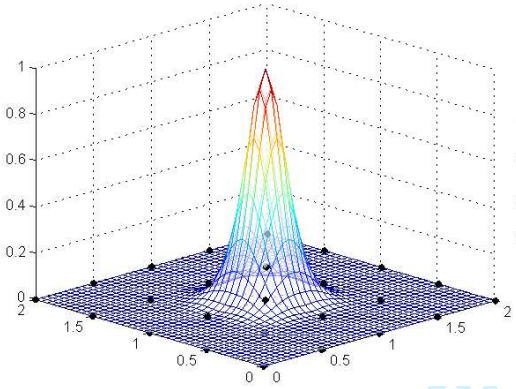
As a further illustration of IMSLS-based shape functions, plots of shape function values over a 2D domain are shown in Figs. 3 and 4. Twenty-five nodes are arranged in a array of five rows and columns over a 2×2 unit domain and the shape function for the central node (located at (1,1)) is plotted over the domain. We compare the shape functions of the IMSLS with MSLS where it is noted that in the MSLS approximation the shape function does not take nodal values when a cubic spline weight function is used as shown in Figs. 3(a) and 3(b), and takes nodal values only when a singular weight function is used as shown in Figs. 3(c) and 3(d). However, oscillations can be observed around the node in Fig. 3(d) using a linear basis. In the proposed IMSLS approximation, the shape function takes nodal values regardless of the type of the weight function, as shown in Fig. 4. It can be seen that the shape function of the central node takes the value of unity at the node itself and diminishes at all the other nodes. For a linear basis, the oscillatory nature seen with the singular weight function is largely absent as shown in Fig. 4(b). Similar situation can be found for the derivatives of shape functions as compared between the two methods in Figs. 5 and 6 (derivatives plotted only in one direction and are the same for the other due to symmetry). The oscillation in Fig. 3(d) is further amplified in Fig. 5(b) for first order derivative and is largely improved in Fig. 6(b).



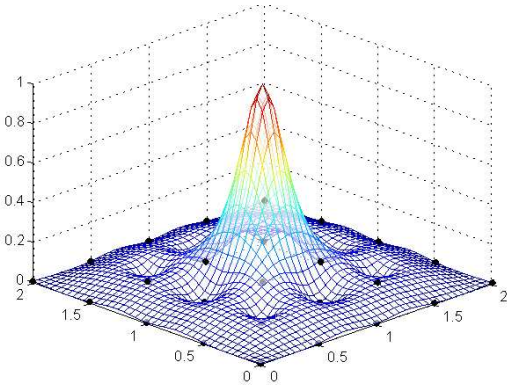
(a) spline weight function and zeroth order basis



(b) spline weight function and linear basis

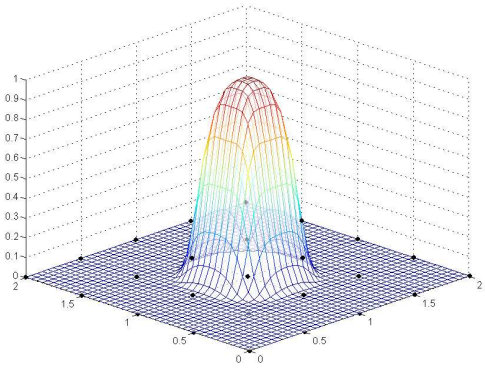


(c) singular weight function and zeroth order basis

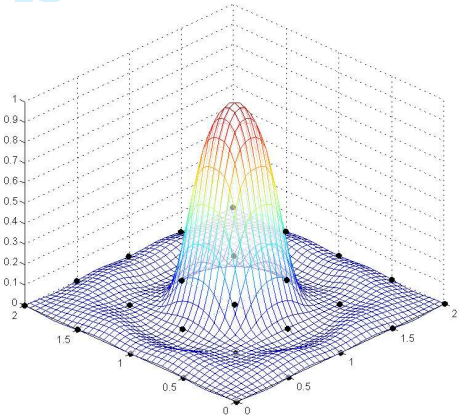


(d) singular weight function and linear basis

Figure 3: 2D plot of MSLS shape functions over a square domain

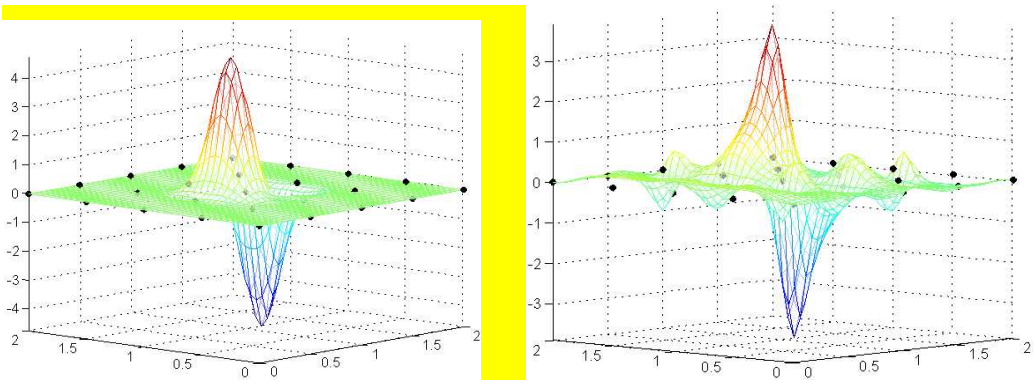


(a) spline weight function and zeroth order basis



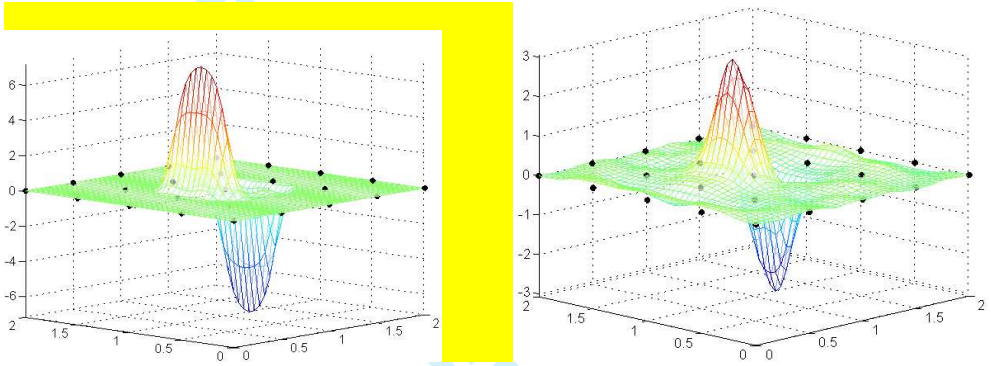
(b) spline weight function and linear basis

Figure 4: 2D plot of IMSLS shape functions over a square domain



(a) singular weight function and zeroth order basis (b) singular weight function and linear basis

Figure 5: 2D plot of MSLS shape functions derivatives over a square domain



(a) spline weight function and zeroth order basis (b) spline weight function and linear basis

Figure 6: 2D plot of IMSLS shape functions derivatives over a square domain

3.4 The derivatives of the IMSLS shape functions and computational cost

In this section, we will firstly show the formulation and properties of the derivatives compare the IMSLS interpolation, and the compare it with the MLS approximation in terms of the computational cost. The IMSLS interpolation at any point $\boldsymbol{x}$ is given by substituting Eq.(3) into (1)

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \left(\sum_{J=1}^{M} \bar{\psi}_J^I(\mathbf{x}) u_J \right) \tag{24}$$

where the definition of $\bar{\psi}_J^I$ is given in Eq. (4) and shape functions are calculated by

$$\psi_J^I(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \mathbf{A}^{-1} \mathbf{B}_J \tag{25}$$

The derivatives of the approximation in Eq.(24) can be obtained by the chain rule

$$u^h(\mathbf{x})_{,k} = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \left(\sum_{J=1}^{M} \bar{\psi}_{J,k}^I(\mathbf{x}) u_J \right) + \sum_{I=1}^{n} \phi_{I,k}(\mathbf{x}) \left(\sum_{J=1}^{M} \bar{\psi}_J^I(\mathbf{x}) u_J \right) \tag{26}$$

where k indicates the derivatives with respect to the k-th coordinate. The derivatives of $\phi_I(\mathbf{x})$ is calculated by

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

$$\phi_{I,k}(\mathbf{x}) = \frac{w_{I,k}(\mathbf{x})\sum_{J=1}^{n} w_J(\mathbf{x}) - w_I(\mathbf{x})\sum_{J=1}^{n} w_{J,k}(\mathbf{x})}{\left(\sum_{J=1}^{n} w_J(\mathbf{x})\right)^2} \tag{27}$$

It can be easily seen that the summation of the PU function derivatives is

$$\sum_{I=1}^{n}\phi_{I,k}(\mathbf{x}) = \frac{\sum_{I=1}^{n} w_{I,k}(\mathbf{x})\sum_{J=1}^{n} w_J(\mathbf{x}) - \sum_{I=1}^{n} w_I(\mathbf{x})\sum_{J=1}^{n} w_{J,k}(\mathbf{x})}{\left(\sum_{J=1}^{n} w_J(\mathbf{x})\right)^2} \equiv 0 \tag{28}$$

And the derivatives of the shape functions by

$$\psi^{I}_{J,k}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}_{,k}(\mathbf{x})\mathbf{A}^{-1}\mathbf{B}_J\,. \tag{29}$$

Denote the final form of the shape functions for the nodes both global and local associated with $\mathbf{x}$ as $N_I(\mathbf{x})$ (see § 4 for the matrix notation of shape functions as a result of global PU multiplying over local approximation). Then Eq. (24) can be rewritten as

$$u^h(\mathbf{x}) = \sum_{I=1}^{R} N_I(\mathbf{x})\, u_I \tag{30}$$

where R is total number of nodes associated with a given point $\mathbf{x}$. Since it has been proved the completeness of the IMSLS shape functions in § 3.2, it can be directly obtained that

$$\sum_{I=1}^{R} N_I(\mathbf{x})\, x_I = x \tag{31}$$

Therefore, the property of the shape function derivatives is

$$\sum_{I=1}^{R} N_I(\mathbf{x})_{,k}\, x_I^j = \delta_{kj} \tag{32}$$

where k denotes the derivatives of the shape function with respect to k-th coordinate of $\mathbf{x}$, and j indicates the j-th component of the coordinates of node I. For example,

$$\sum_{I=1}^{R} N_I(\mathbf{x})_{,1}\, x_I^1 = \sum_{I=1}^{R}\left(\frac{\partial N_I}{\partial x} x_I\right) = 1\,, \tag{33}$$

and

$$\sum_{I=1}^{R} N_I(\mathbf{x})_{,2}\, x_I^1 = \sum_{I=1}^{R}\left(\frac{\partial N_I}{\partial y} x_I\right) = 0\,. \tag{34}$$

To make comparisons of the computational cost, the formulation of the MLS approximation [17] is briefly stated in the following. Terms in the MLS approximation similar to the present IMSLS interpolation will be marked with a tilde, i.e. $\tilde{\mathbf{A}}$ and $\mathbf{A}$ etc.

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

In the MLS approximation, field variables, such as displacement in solid mechanics, are also approximated with shape functions over nodal values as

$$u(\mathbf{x}) = \sum_{I=1}^{n} \Phi_I(\mathbf{x}) u_I \tag{35}$$

where Φ_I are the MLS shape functions, computed by

$$\Phi_I(\mathbf{x}) = \sum_k p_j(\mathbf{x}) \left(\tilde{\mathbf{A}}^{-1}(\mathbf{x}) \tilde{\mathbf{B}}(\mathbf{x}) \right)_{jI} = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \tilde{\mathbf{A}}^{-1}(\mathbf{x}) \tilde{\mathbf{B}}_I(\mathbf{x}), \tag{36}$$

and the matrix $\tilde{\mathbf{A}}(\mathbf{x})$ is given by

$$\tilde{\mathbf{A}}(\mathbf{x}) = \sum_{I=1}^{n} w_I(\mathbf{x}) \mathbf{p}(\mathbf{x}_I) \mathbf{p}^{\mathrm{T}}(\mathbf{x}_I) \tag{37}$$

and the matrix $\tilde{\mathbf{B}}(\mathbf{x})$ by

$$\tilde{\mathbf{B}}(\mathbf{x}) = \left[w_1(\mathbf{x}) \mathbf{p}(\mathbf{x}_1) \quad w_2(\mathbf{x}) \mathbf{p}(\mathbf{x}_2) \quad \cdots \quad w_n(\mathbf{x}) \mathbf{p}(\mathbf{x}_n) \right]. \tag{38}$$

The derivatives of the shape functions can be found by applying the chain rule to Eq. (36)

$$\Phi_{I,k} = \mathbf{p}_{,k}^{\mathrm{T}} \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{B}}_I - \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{A}}_{,k} \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{B}}_I + \mathbf{p}^{\mathrm{T}} \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{B}}_{I,k}, \tag{39}$$

where the definition of k is same as in Eq. (26). Eqs. (37) and (38) show the dependence of $\tilde{\mathbf{A}}(\mathbf{x})$ and $\tilde{\mathbf{B}}(\mathbf{x})$ on $\mathbf{x}$ respectively, which then needs to be differentiated with respect to $\mathbf{x}$ as shown in Eq. (39). For the IMSLS interpolation, $\mathbf{A}$ and $\mathbf{B}$ in Eq. (25) only depend on the coordinates of nodes in support, and thus neither needs to be differentiated in Eq. (29). By comparing Eq. (39) with (29), it can be seen that the IMSLS interpolation has a more compact formulation and involves many fewer matrix operations, which can only mean a reduced computational cost. Therefore, the IMSLS works more efficiently than the MLS at each interpolation point. However, this is not yet sufficient evidence to assert that the total computing time of the IMSLS is less than the MLS for any given problem since the total computing time depends on both the computing time in each interpolation and the total number of interpolations required. And it is therefore necessary to compare the number of interpolations that need to be performed for a certain problem. In the IMSLS interpolation, the matrix inversion appears only in the local approximation at each node, which means the inversion of $\mathbf{A}$ is required only once for each node, thus the total number of matrix inversions should be the same as the number of nodes. With the MLS approximation, matrix inversion is performed at each integration point. Generally for a certain problem the number of integration points needed is much

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

greater than the number of nodes in the MLS in order to obtain the weak form integration with satisfactory accuracy. Therefore, it can be seen that the IMSLS surpasses the MLS approximation in terms of number of interpolations required. Therefore, the total computational cost is greatly reduced in the IMSLS. The computational depends on several factors such as the solver either direct or iterative, preconditioning for better convergence regarding the type of the problems, the date storage structure and etc [31]. To show the substantial difference of computational cost, the running time are compared between the two approximations in § 5.1.

4. Discretisation of the weak form

Let R be the total number of nodes associated with a given point $\mathbf{x}$, then Eq. (24) can be rewritten as

$$u^h(\mathbf{x}) = \begin{Bmatrix} \varphi_1^0 \\ \varphi_2^0 \\ \vdots \\ \varphi_n^0 \end{Bmatrix}^{\mathrm{T}} \left[\begin{array}{ccc|ccc} \bar{\psi}_1^1 & \cdots & \bar{\psi}_n^1 & \bar{\psi}_{n+1}^1 & \cdots & \bar{\psi}_R^1 \\ \bar{\psi}_1^2 & \cdots & \bar{\psi}_n^2 & \bar{\psi}_{n+1}^2 & \cdots & \bar{\psi}_R^2 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \bar{\psi}_1^n & \cdots & \bar{\psi}_n^n & \bar{\psi}_{n+1}^n & \cdots & \bar{\psi}_R^n \end{array}\right] \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_R \end{Bmatrix} \tag{40}$$

$$= \underset{1\times n}{\boldsymbol{\Phi}^0}\ \underset{n\times R}{\bar{\boldsymbol{\psi}}}\ \underset{R\times 1}{\mathbf{u}} = \sum_{k=1}^{R} \bar{N}_k(\mathbf{x}) u_k$$

where $\boldsymbol{\Phi}^0$ is the vector of Shepard shape functions, $\bar{\boldsymbol{\psi}}$ is a matrix comprising the point interpolation in Eq. (4) and $\bar{N}_k(\mathbf{x})$ is the IMSLS shape function. In the implementation, all the nodes in local support is normally more than the nodes than in PU support meaning $R \geq n$. For boundary nodes, it is clear than $R = n$.

With the interpolation defined, then the problem domain can be discretised using a weak form, e.g. a Galerkin procedure as used here, and the rest of the implementation is mostly identical to the conventional FEM. We state the discretisation of the weak form for plane stress linear elasticity with small displacements on the domain Ω bounded by Γ, the standard principle of minimum potential energy leads to the following expression:

$$\Pi = \int_\Omega \frac{1}{2} \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} t dx dy - \int_{s_\sigma} \mathbf{u}^T \mathbf{T} t ds - \int_\Omega \mathbf{u}^T \mathbf{b} t dx dy \tag{41}$$

where $\boldsymbol{\varepsilon}$ is the infinitesimal strain vector; $\mathbf{D}$ is the elasticity matrix; $\mathbf{T}$ is the surface force vector; $\mathbf{b}$ is the body force vector and t is the thickness of the two-dimensional body. If we substitute Eq. (40) into (41) and invoke $\delta\Pi = 0$, we will get the following discrete equation

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

$$\mathbf{K}\cdot\mathbf{U}=\mathbf{F} \tag{42}$$

where the stiffness matrix is

$$\mathbf{K}_{IJ}=\int_{\Omega}\bar{\mathbf{B}}_I^{\mathrm{T}}\cdot\mathbf{D}\cdot\bar{\mathbf{B}}_J\mathbf{d}\Omega \tag{43}$$

in which $\bar{\mathbf{B}}_I$ is the strain-displacement matrix

$$\bar{\mathbf{B}}_I=\begin{bmatrix}\bar{N}_{I,x} & 0\\ 0 & \bar{N}_{I,y}\\ \bar{N}_{I,y} & \bar{N}_{I,x}\end{bmatrix} \tag{44}$$

and $\mathbf{F}_i$ is the right hand side vector

$$\mathbf{F}_i=\int_{S_\sigma}\bar{N}_i\cdot\mathbf{T}tds+\int_{\Omega}\bar{N}_i\cdot\mathbf{b}d\Omega\,. \tag{45}$$

Eq. (41) can be integrated by Gaussian integration scheme using background integration cells. A Delaunay triangulation can be generated for this purpose from the nodes of the meshless model with four integration points in each triangle.

5. Numerical examples

The proposed improved IMSLS interpolation has been coded into an existing C++ program. In this section, we show the performance of the interpolation on a range of test problems. Results obtained are compared with the exact solutions, those given by the EFG method and also the FEM . The weight function used in the EFG method for testing purposes is the exponential weight function given by

$$w_I^c(\mathbf{x})=\begin{cases}\dfrac{e^{-(r_I/c_I)^2}-e^{-(r_{II}/c_I)^2}}{1-e^{-(r_{II}/c_I)^2}}, & \text{if } r_i\le r_{Ii}\\ 0, & \text{if } r_i>r_{Ii}\end{cases} \tag{46}$$

where r_{II} is defined by Eq.(9) and $c_I=0.3r_{II}$ is used for all test examples. Unless otherwise indicated, the scale factor a in Eq. (9) is set as 1.5 in the EFG method and as 1.1 in the IMSLS. The same integration schemes are kept in both the proposed method and the EFG method. To study the convergence behaviour we define the following error norms in displacement and energy respectively

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

$$\|\mathbf{u}\| = \left(\int_{\Omega} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{u} d\Omega \right)^{\frac{1}{2}}, \tag{47}$$

where $\mathbf{u}$ is a vector collecting nodal displacement results $\mathbf{u} = \{u_1, v_1, u_2, v_2 \cdots u_n, v_n\}^T$ and

$$\|\boldsymbol{\varepsilon}\| = \left(\int_{\Omega} \boldsymbol{\varepsilon}^{\mathrm{T}} \cdot \boldsymbol{\sigma} d\Omega \right)^{\frac{1}{2}} \tag{48}$$

where $\boldsymbol{\varepsilon}$ is the infinitesimal strain vector and $\boldsymbol{\sigma}$ is the Cauchy stress vector. The relative displacement error and energy error are given by

$$r_u = \frac{\left\| \mathbf{u}^{\mathrm{num}} - \mathbf{u}^{\mathrm{exact}} \right\|}{\left\| \mathbf{u}^{\mathrm{exact}} \right\|} \tag{49}$$

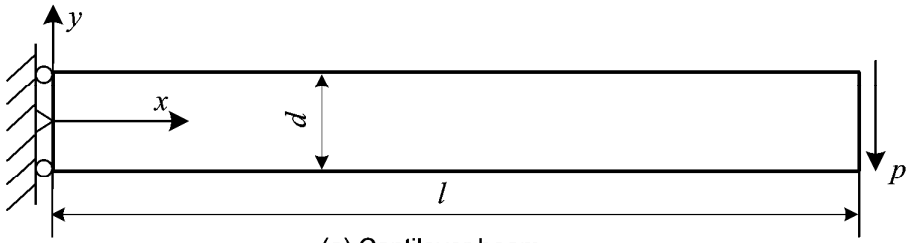
and

$$r_e = \frac{\left\| \boldsymbol{\varepsilon}^{\mathrm{num}} - \boldsymbol{\varepsilon}^{\mathrm{exact}} \right\|}{\left\| \boldsymbol{\varepsilon}^{\mathrm{exact}} \right\|} \tag{50}$$

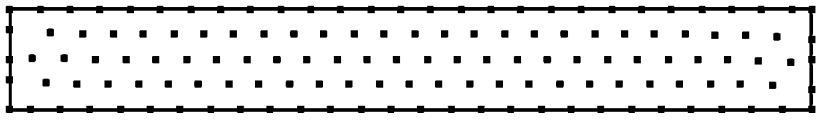
where the superscripts *num* and *exact* refer to numerical solutions and exact (or reference) solutions respectively.

5.1 A cantilever beam

A cantilever beam problem with dimensions of l = 8 m and d = 1m, as shown in Fig. 7 is tested first. The beam is subjected to a unit concentrated load p at the right-hand end and is constrained at the left-hand end as shown in the Figure. The elastic material properties used are $E = 1 \times 10^5 \, Pa$ and $\nu = 0.25$ and the problem is solved under a plane strain assumption. We refer to the analytical solution of the problem given in [33]. The convergence of the present method is studied using four nodal arrangements with 50, 138, 486 and 965 nodes respectively.



(a) Cantilever beam



(b) Nodal arrangement

Figure 7: Cantilever beam and nodal arrangement

The convergence rate is compared between FEM using three-noded triangles, the EFG method and the IMSLS in Figs. 8 and 9. It can be seen that the IMSLS shows good accuracy and convergence rate. Figs. 10 and 11 collect the vertical displacement v and σ_{xx} along $y = d/2$ on the beam by FEM, EFG method and IMSLS using a nodal arrangement of 138 nodes, and also indicates the good accuracy of results using the proposed formulation. Note that in this example, the reason of using the triangular element in the FEM as comparison is that the three-noded triangular elements is constructed from a linear basis in 2D $\{1, x, y\}$ which is corresponding to the linear basis in the IMSLS. While for higher order element, e.g. the quadrilateral element, the shape function corresponds to a bilinear basis $\{1, x, y, xy\}$ which is of higher order than the linear basis in the IMSLS. The choice of element is therefore based on the same order of basis function employed for expediency though the triangular element is known poor performance in cantilever beam.

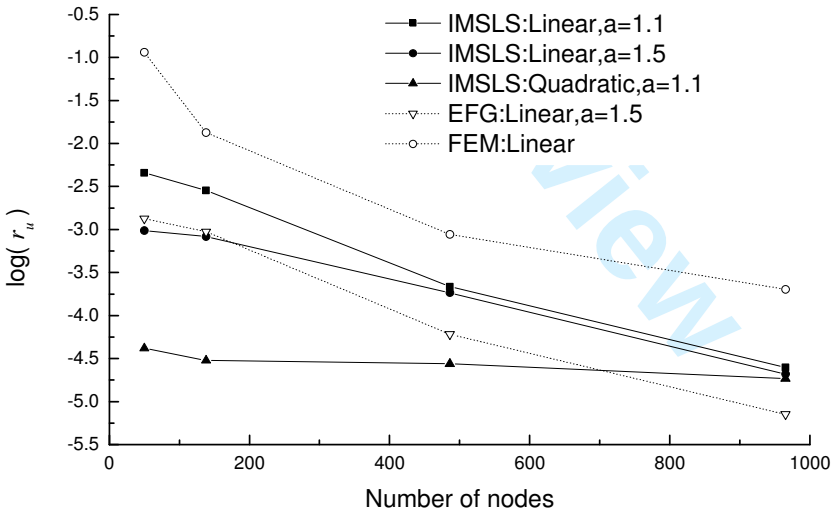


Figure 8: Convergence of relative displacement error of the cantilever beam

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

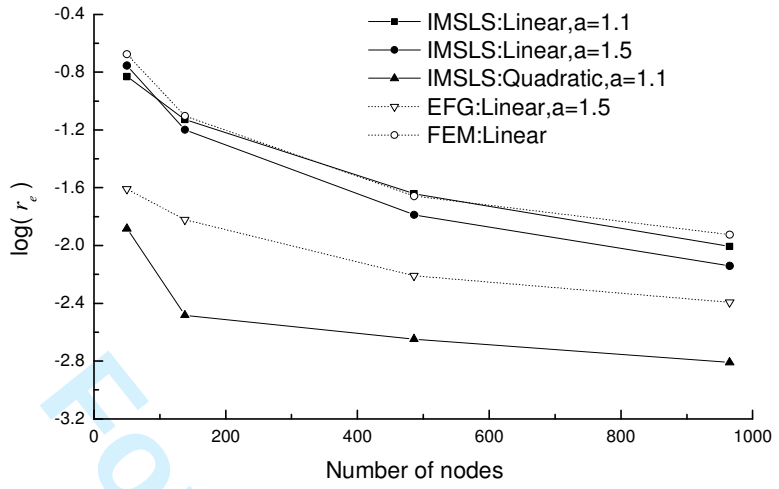


Figure 9: Convergence of relative energy error of the cantilever beam

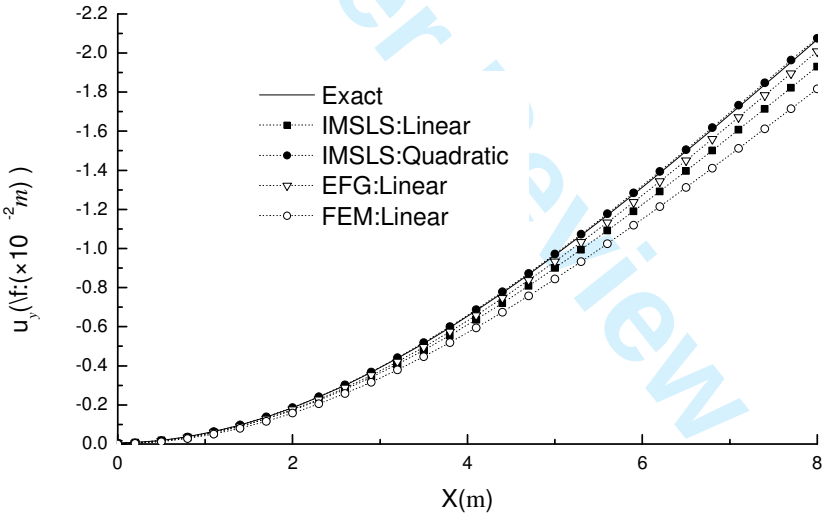


Figure 10: Vertical displacement results v along $y = 0$ of the cantilever beam

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

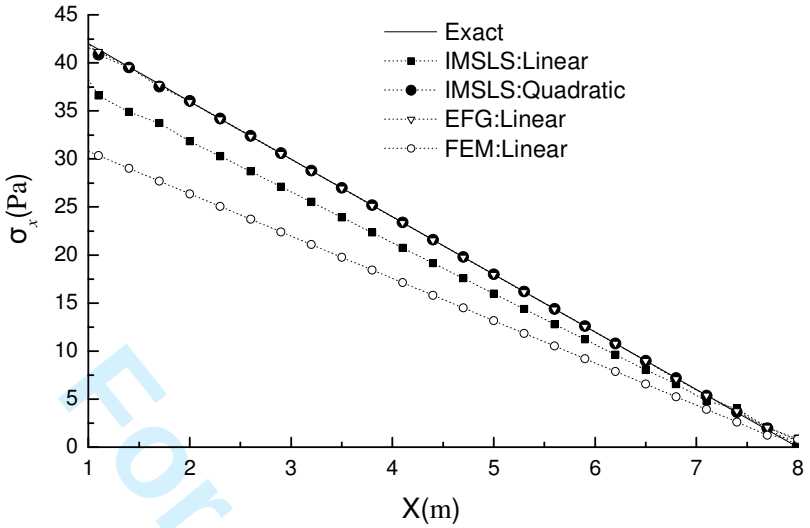


Figure 11: σ_{xx} results along $y = 0$ of the cantilever beam

Table 1: Comparison of computing time in obtaining the strain matrix (unit: second)

Number of nodes	50	138	486	965
MLS	0.16	0.59	2.31	4.35
IMSLS	0.14	0.46	1.41	2.50

Figs. 8 and 9 also demonstrate that the present interpolation is slightly improved with an increase of the size of support domain for local approximation. The issue of optimum nodal support size with respect to error control has been found with the EFG method [33] and the similar issue here can be discussed in further study. It should be noted that in the present exapmle, the EFG method outperformed all the other methods using linear basis, only the quadratic basis of the present IMSLS performs better. As has been highlighted in Section 3, the computational cost in obtaining the shape functions and its derivatives is much lower by the present LS interpolation than by the MLS approximation. And this point is here clearly proved by the computational time in obtaining the strain matrix listed in Tab.1. It can be observed from the table that the difference in computational efficiency between the two interpolations increases when the number of nodes increases.

5.2 An infinite plate with a circular hole

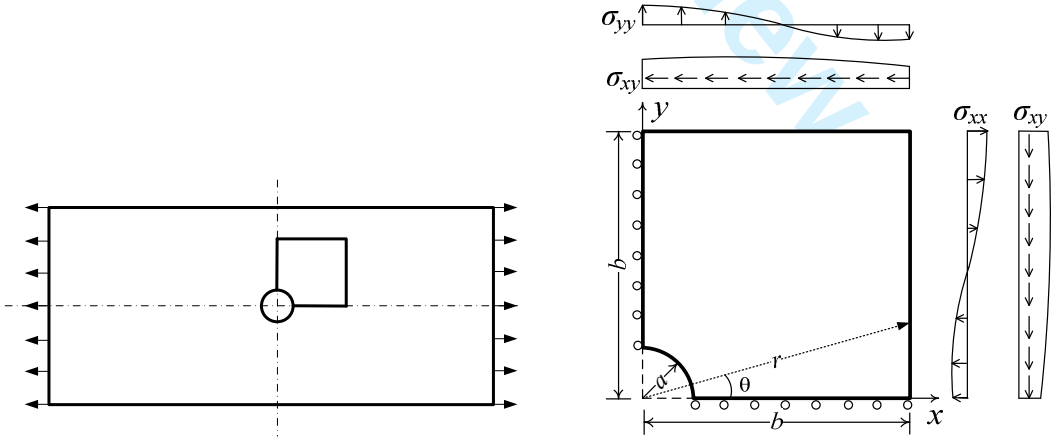
The second example is an infinite plate with a circular hole of radius a = 1 m. The plate is subjected to far field traction $\sigma = 1\,Pa$ in the x direction. A finite portion of the plate is considered for analysis and, due to the symmetry of the problem, only a quarter of the portion requires modeling, as shown in Fig. 12. The elastic material properties used are $E = 3.0\times10^{7}\,Pa$ and $\nu = 0.3$ and plane stress conditions are assumed. The stresses and displacements for this are given in an analytical solution in [32] as

$$\begin{aligned}
\sigma_{xx} &= 1-\frac{a^2}{r^2}\left(\frac{3}{2}\cos(2\theta)+\cos(4\theta)\right)+\frac{3a^4}{2r^4}\cos(4\theta)\\
\sigma_{xy} &= -\frac{a^2}{r^2}\left(\frac{1}{2}\sin(2\theta)+\sin(4\theta)\right)+\frac{3a^4}{2r^4}\sin(4\theta)\\
\sigma_{yy} &= -\frac{a^2}{r^2}\left(\frac{1}{2}\cos(2\theta)-\cos(4\theta)\right)-\frac{3a^4}{2r^4}\cos(4\theta)
\end{aligned} \tag{51}$$

and

$$\begin{aligned}
u &= \frac{a}{8G}\left(\frac{r}{a}(\kappa+1)\cos(\theta)+\frac{2a}{r}\left[(1+\kappa)\cos(\theta)+\cos 3(\theta)\right]-\frac{2a^3}{r^3}\cos 3(\theta)\right)\\
v &= \frac{a}{8G}\left(\frac{r}{a}(\kappa-3)\sin(\theta)+\frac{2a}{r}\left[(1-\kappa)\sin(\theta)+\sin 3(\theta)\right]-\frac{2a^3}{r^3}\sin 3(\theta)\right)
\end{aligned} \tag{52}$$

where G is the shear modulus and κ is the Kolosov constant where $\kappa = (3-\nu)/(1-\nu)$ for the plane strain assumption.



(a) A small portion taken for analysis (b) Boundary conditions applied

Figure 12: An infinite plate with a circular hole

Traction-prescribed boundary conditions consistent with the exact solution in Eq. (42) are applied at the top and right edges. Four distributions of 53, 188, 564 and 1012

nodes, which are shown in Fig.13, are employed for the convergence studies. Figs.14 and 15 show that the IMSLS has a good convergence performance in the displacement and energy norm. For the relative error in displacement, the error of IMSLS is between the EFG and FEM given the same node density. In this example the EFG method outperformed the IMSLS linear methods. The displacement u_x obtained using the IMSLS and the EFG method are shown in Fig.16.

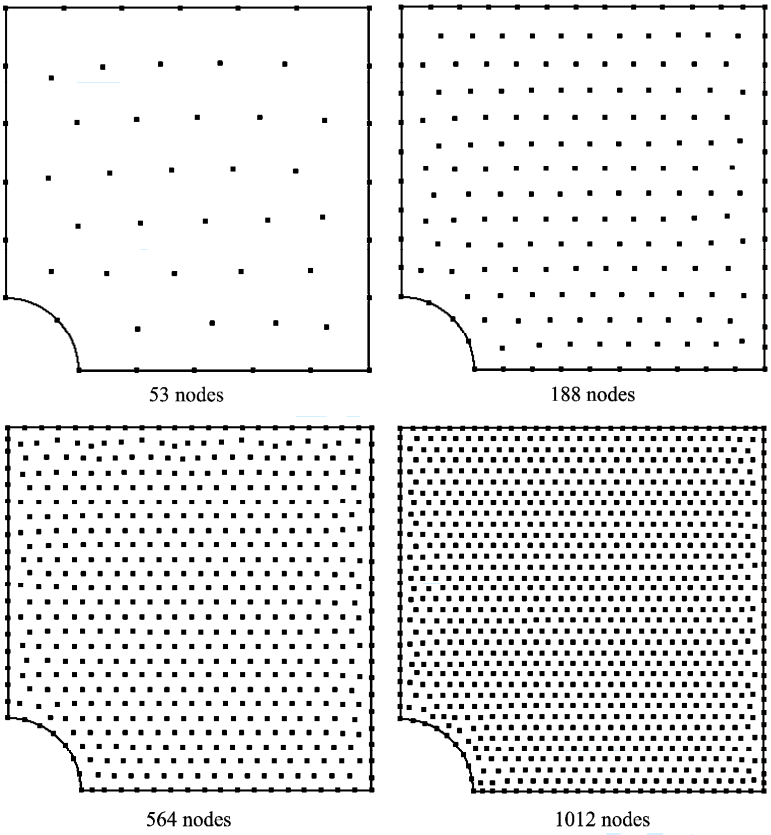


Figure 13: Nodal arrangements used for the infinite plate problem

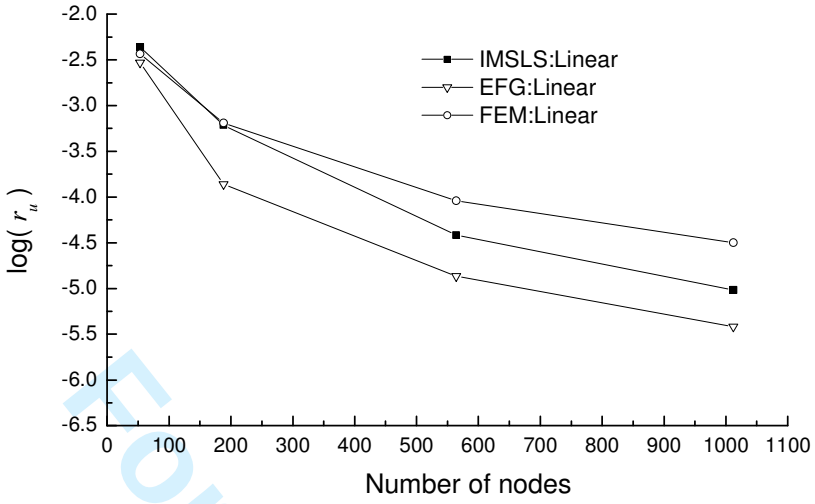


Figure 14: Convergence of relative displacement error for the infinite plate problem

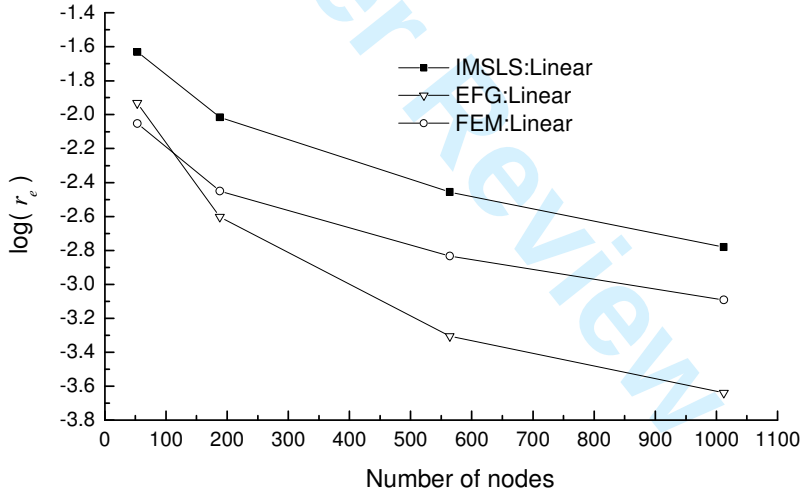


Figure 15: Convergence of relative energy error for the infinite plate problem.

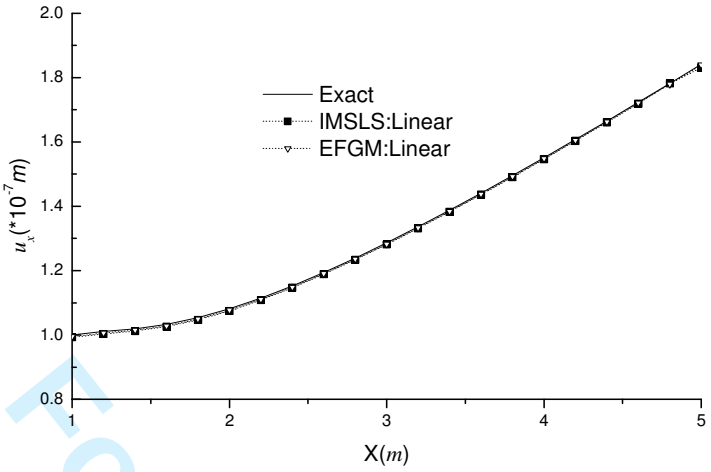


Figure 16: Comparison of the horizontal displacement u along $y = 0$ by different methods

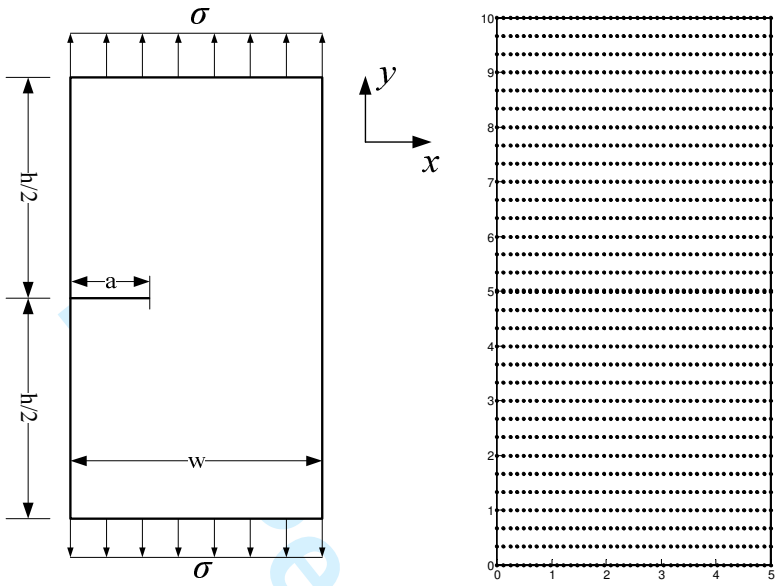
5.3 A square plate with an edge crack

The last test example is a rectangular plate with a single edge crack. The dimensions of the plate used in the test are L = 10m and W = 5m as shown in Fig. 17. The plate is subjected to uniform traction of σ = 1 in the y direction. Boundary conditions are applied as shown in Fig. 17 (a). The elastic material parameters used are $E = 3.0\times10^7$ and $\upsilon = 0.3$ and the problem is solved under plane strain assumption. A linear basis in 2D is used in this example. A structured nodal arrangement of 1344 nodes is used as shown in Fig. 17 (b). We test this example by varying the length of crack and study the accuracy via the stress intensity factor (SIF) as the fundamental fracture parameter. SIFs are used both to indicate stability, i.e. likelihood of propagation, and to determine the direction of crack growth with respect to the current geometry. The SIF is here computed using the J-integral [34] using the stress and strain results obtained. For linear elastostatics, without body forces and assuming traction free states along crack surfaces, the J integral defines the energy release rate along a path as

$$J_k = \int_\Lambda W n_k - t_j u_{j,k} \mathrm{d}\Lambda \qquad j,k = 1,2 \tag{53}$$

where W is the strain energy calculated by $W = \frac{1}{2}\sigma_{ij}\varepsilon_{ij}$, t_j is the traction along Λ calculated by $t_j = \sigma_{ji}n_i$, and $u_{j,k}$ is the derivatives of the j-th component of displacement with respect to the kth axis. Here j and k indicate the local coordiantes

defined around the crack tip. In linear elastic fracture mechanics, J_1 is normally used since it does not contain singular terms and J_1 can be decomposed into symmetric and anti-symmetric parts as described in [35].



(a) dimensions and boundary conditions (b) nodal arrangement

Figure 17: A single edge crack in a square plate

In Table 2, we compare the values of normalized SIF ($F_I = K_I / \sigma\sqrt{\pi a}$) obtained by the present method, the original MSLS method using a singular weight function and the EFG method with the reference values in [36]. The results show an improvement of accuracy with the present IMSLS method compared to the original MSLS method. The results also indicate the EFG method using the visibility criterion leads to significant errors which is due to the spurious crack extension problem in the MLS approximation as has been reported in [37]. It also shows that with the same number of nodes used, the IMSLS performs much better than the widely used MLS approximation, and the total computational cost is much lower by the former as is explained in § 3.4.

Table 2: Normalized SIF results for the single edge crack problem

a/W	0.2		0.4		0.6	
	F_I	Error(%)	F_I	Error(%)	F_I	Error(%)
Reference value	1.370	--	2.110	--	4.030	--
IMSLS	1.343	-1.97	2.078	-1.50	3.981	-1.201
MSLS	1.320	-3.64	2.009	-4.78	3.757	-6.77
EFG	1.476	7.74	2.233	5.83	4.313	7.03

6. Conclusions

In this paper we propose an improved meshless Shepard and least square interpolation which removes the drawbacks associated with the use of singular weight function in the original MSLS method. The support domain for constructing local approximation and the support domain for PU approximation are dually defined at each node, which delivers the ideal delta property along essential boundaries without using singular weight functions. The present interpolation benefits from a simple formulation of shape functions and their derivatives, which makes it easier to implement than the MLS approximation. In addition the computational cost in obtaining shape functions is much lower than using the MLS approximation. The features of the proposed IMSLS interpolation can be summarised as follows

(1) The present interpolation satisfies the delta property on essential boundaries without using singular weight functions so that essential boundary conditions can be imposed as directly as in the FEM;

(2) The computational cost in obtaining the shape functions and their derivatives is much lower than the widely used MLS approximation;

(3) The proposed interpolation preserves the consistency up to the order of the basis function, which is a necessary requirement of accuracy;

(4) The proposed interpolation starts to converge towards the real solution even with a small size of support domain and such convergence characteristic is not sensitive when the size of support domain increases.

Based on the above described advantages, which are the necessary elements to make a meshless method useful for application and which are absent from many other meshless methods, we conclude that the proposed IMSLS interpolation is a promising meshless method worthy of consideration in a variety of applications. The formulation here is derived for 2D analysis but is readily extendable to 3D and the essential ideas are the same. Further development of the proposed interpolation is ongoing with its application to problems of changing geometry, such as those including finite deformation, elastoplasticity and three-dimensional cracking problems.

Acknowledgements

The authors gratefully acknowledge the support of Natural Science Foundation of China (NSFC 41130751), National Basic Research Program of China (973 Program:

2011CB013800), Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT, IRT1029), Shanghai Pujiang Talent Program (12PJ1409100) and Shanghai Chenguang Talent Program (12CG20).

References

1. Belytschko T, Krongauz Y, Organ D. Meshless methods: An overview and recent developments. Computer Methods in Applied Mechanics and Engineering 1996;139: 3-47
2. Fries T, Matthies, H. Classification and overview of Meshfree Methods. Technical report. Technical University Braunschweig, Germany, 2004.
3. Nguyen, V.; Rabczuk, T.; Bordas, S. & Duflot, M.Meshless methods: A review and computer implementation aspects. Mathematics and Computers in Simulation 2008; 79: 763 – 813.
4. Zhang Z, Liew KM, Cheng YM, Li YY. Analyzing 2D fracture problems with the improved element-free Galerkin method. Engineering Analysis with Boundary Elements 2008; 32: 241-256.
5. Liew KM, Cheng YM, Kitipornchai S. Analyzing the 2D fracture problems via the enriched boundary element-free method. International Journal of Solids and Structures2007; 44: 4220-4233.
6. Long SY, Liu KY, Li GY. An analysis for the elasto-plastic fracture problem by the meshless local Petrov-Galerkin method. Computer Modeling in Engineering and Sciences (CMES), 2008; 28: 203-216.
7. Krysl P, Belytschko T. Analysis of thin shells by the element-free Galerkin method. International Journal for Numerical Methods in Engineering 1996; 33: 3057-3078.
8. Chen JS, Pan C, Wu CT, Liu WK. Reproducing Kernel Particle Methods for Large Deformation Analysis of Nonlinear Structures. Computer Methods in Applied Mechanics and Engineering 1996; 139: 195-227.
9. Heaney CE, Augarde CE, Deeks AJ. Modelling elasto-plasticity using the hybrid MLPG method. Computer Modeling in Engineering and Sciences 2010; 56: 153-178.
10. Wang JG, Liu GR, Lin P. Numerical analysis of Biot's consolidation process by radial point interpolation method. International Journal of Solids and Structures 2002; 39:1557-1573.
11. Li S, Liu W, Rosakis A, Belytschko T, Hao W. Mesh-free Galerkin simulations of dynamic shear band propagation and failure mode transition. International Journal of Solids and Structure, 2002; 39: 1213-1240.
12. Han ZD, Atluri SN. Meshless Local Petrov–Galerkin (MLPG) approaches for solving 3D problems in elasto-statics. Computer Modeling in Engineering and Sciences (CMES) 2004; 6: 169-188.
13. Zhuang X, Augarde C, Mathisen K. Fracture modelling using meshless methods and level sets in 3D: framework and modelling. International Journal for Numerical Methods in Engineering 2012; 92: 969-998.
14. Liu WK, Han W, Lu H, Li S. Reproducing Kernel Element Method, Part I Theoretical Formulation Computer Methods in Applied Mechanics and Engineering, 193: 933–951, 2004.
15. You Y, Chen JS, Lu H. Filter. Reproducing Kernel, and Adaptive Meshfree Methods. Computational Mechanics, 31: 316-326, 2003.
16. Wu CT, Park CK, Chen JS. A generalized approximation for the meshfree analysis of solids," International Journal for Numerical Methods in Engineering. 85: 693-722, 2011.
17. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. Mathematics of Computation 1981; 37: 141-158.
18. Breitkopf P, Rassine A, Touzot G. Explicit form and efficient computation of MLS shape functions and their derivatives. International Journal for Numerical Methods in Engineering, 2000; 48:451-466.

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany

19. Zhuang X, Augarde C. Aspects of the use of orthogonal basis functions in the element free Galerkin method. International Journal for Numerical Methods in Engineering, 2010; 81: 366-380.
20. Kaljevic I, Saigal AS. An improved element free Galerkin formulation. International Journal for Numerical Methods in Engineering, 1997; 40: 2953-2974.
21. Atluri, S.N.; Kim, H.G.; Cho, J.Y. A critical assessment of the truly meshless local Petrov-Galerkin and local boundary integral equation methods. Computational Mechanics 1999; 24: 348–372.
22. Krongauz Y, Belytschko T. Consistent pseudo-derivatives in meshless methods. Computer Methods in Applied Mechanics and Engineering 1997; 146: 371-386.
23. Macri M, De S. An octree partition of unity method (OctPUM) with enrichments for multiscale modeling of heterogeneous media. Computers and Structures 2008; 86: 780-795.
24. Griebel M, Schweitzer MA. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. SIAM Journal on Scientific Computing 2000; 22: 853-890.
25. Griebel M, Schweitzer MA. A particle-partition of unity method-Part II: Efficient cover construction and reliable integration. SIAM Journal on Scientific Computing 2002; 23: 1665-1682.
26. Griebel M, Schweitzer MA. A particle-partition of unity method-Part III: A multilevel solver. SIAM Journal on Scientific Computing 2002; 24: 377-409.
27. Cai YC, Zhu HH. A local meshless Shepard and least square interpolation method based on local weak form. Computer Modeling in Engineering & Sciences, 2008; 34: 179-204.
28. Cai YC, Zhu HH. A PU-based meshless Shepard interpolation method satisfying delta property. Engineering Analysis with Boundary Elements 2010; 34: 9-16.
29. Oden JT, Durate CA, Zienkiewicz OC. A new cloud-based hp finite element method. Computer Methods in Applied Mechanics and Engineering 1998; 153:117-126.
30. Liu GR, Gu YT, Dai KY. Assessment and applications of point interpolation methods for computational mechanics. International Journal for Numerical Methods in Engineering 2004; 59: 1373-1397.
31. Collier C, Pardo D, Dalcin L, Paszynski M, Calo VM. The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. Computer Methods in Applied Mechanics and Engineering 2012; 213-216:353 -361.
32. Belytschko T, Lu YY, Gu L. Element-free Galerkin method. International Journal for Numerical Methods in Engineering 1994; 37: 229-256.
33. Zhuang X, Heaney C, Augarde C. On error control in the element-free Galerkin method. Engineering Analysis with Boundary Elements 2012; 36:351-360.
34. Rice J. A path independent integral and the approximate analysis of strain concentration by notches and cracks. Journal of Applied Mechanics 1968; 35:379-386.
35. Portela A, Aliabadi M, Rooke D. The dual boundary element method—effective implementation for crack problems. International Journal for Numerical Methods in Engineering 1992; 33:1269-1287.
36. Murakami Y. The stress intensity factors handbook. Pergamon Press: Oxford, 1987.
37. Zhuang X, Augarde C. and Bordas S. Accurate fracture modelling using meshless methods and level sets: formulation and 2D modelling. International Journal for Numerical Methods in Engineering 2011; 86:249-268.

Prof. Dr.-Ing. P. Wriggers, Institute of Continuum Mechanics, Appelstr. 11, 30167 Hannover, Germany